

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS ARARANGUÁ**

Diogo Rodrigues de Souza

**IMPLANTAÇÃO DA METODOLOGIA ÁGIL SCRUM EM  
UM AMBIENTE DE DESENVOLVIMENTO**

Araranguá, dezembro de 2014.

Diogo Rodrigues de Souza

# **IMPLANTAÇÃO DA METODOLOGIA ÁGIL SCRUM EM UM AMBIENTE DE DESENVOLVIMENTO**

Trabalho de Conclusão de Curso  
submetido à Universidade Federal de  
Santa Catarina, como parte dos  
requisitos necessários para a obtenção  
do Grau de Bacharel em Tecnologias  
da Informação e Comunicação.

Orientadora: Prof<sup>ª</sup>. Dr.<sup>a</sup> Luciana Bolan  
Frigo

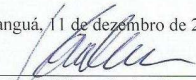
Araranguá, dezembro de 2014.

Diogo Rodrigues de Souza

## IMPLANTAÇÃO DA METODOLOGIA ÁGIL SCRUM EM UM AMBIENTE DE DESENVOLVIMENTO

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de Bacharel em Tecnologias da Informação e Comunicação, e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 11 de dezembro de 2014.



---


Prof. Vilson Gruber Dr.  
Coordenador do Curso

### **Banca Examinadora:**



---

Prof.ª Luciana Bolan Frigo, Dr.ª  
Orientadora



---

Prof.ª Analucia Schiaffino Morales, Dr.ª



---

Prof. Gustavo Medeiros, Dr.

Dedico este trabalho a todos que estiveram a minha volta durante o desenvolvimento do mesmo, em especial os meus pais, meus irmãos, minha namorada, e colegas de trabalho.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, não pela conclusão do presente estudo, mais sim pela oportunidade de acordar todos os dias e realizar tudo que venho buscando.

A minha professora-orientadora Dr.<sup>a</sup> Luciana pela dedicação e por todo o conhecimento que me foi passado, tornando a realização deste possível. Como também, a todos os professores que me auxiliaram no decorrer da graduação, vocês têm um papel fundamental nos meus próximos passos.

A meus pais, não só pelo apoio no decorrer deste estudo, mais sim pelo apoio em todas as etapas da minha vida. A meus irmãos, estando sempre prontos para me atender nos momentos de insatisfação e compartilhar comigo os momentos de alegria, em especial a minha irmã Fabíola pelo “socorro” prestado em dúvidas com o português. A minha namorada, pela compreensão e pelo apoio incondicional mesmo nos momentos em que não pude corresponder da mesma forma.

A empresa e aos amigos que lá fiz, pois me proporcionaram um ótimo ambiente de trabalho, permitindo testar alguns conceitos e desenvolver esse projeto, assim como também agradecer o aprendizado obtido a cada dia.

Por fim, meus sinceros agradecimentos a todos que de alguma forma vem contribuindo para a minha formação.

Concentre-se nos pontos fortes, reconheça as fraquezas, agarre as oportunidades e proteja-se contra as ameaças (Sun Tzu, 500 a.C.).

## RESUMO

As metodologias ágeis de desenvolvimento de software têm tido grande aceitação no Brasil, pois servem como ferramenta na busca por melhores resultados em produtos de software, fornecendo aprimoramento de gerenciamento e de processos utilizados nesses projetos. Mesmo conhecendo os benefícios que estas metodologias podem trazer, o processo de implantação nem sempre é simples, pois não depende só de uma alteração estrutural da empresa, mas também na cultura organizacional da mesma. O objetivo deste estudo é apresentar uma experiência de implantação da metodologia ágil Scrum. Essa metodologia se destacou pois visa a realização de incrementos que agreguem mais valor ao negócio, podendo assim trazer uma série de melhorias, como também algumas perdas, uma vez que proporciona uma reflexão organizacional com várias mudanças, e não só a nível de estrutura de desenvolvimento de código. Antes da implantação do Scrum não havia uso específico de uma metodologia para apoiar o gerenciamento do processo de software, o que foi um fator relevante para a busca de uma mudança. A adoção do Scrum foi um processo necessário para a melhoria da empresa, despertando o interesse da equipe e melhorando os processos organizacionais, o que tornou a equipe ágil e permitindo um melhor conhecimento dos resultados obtidos.

**Palavras-chave:** Métodos ágeis. Engenharia de Software. Cultura ágil. Scrum.

## **ABSTRACT**

The agile software development have been widely accepted in Brazil because they serve as a tool in the search for better results in software products, providing improved management and processes used in these projects. Even knowing the benefits that these methodologies can bring, the deployment process is not always simple, as it not only depends on a structural change of the company but also in the organizational culture of the same. The objective of this study is to present a deployment experience of Scrum agile methodology. This methodology stood out because it seeks the realization of increments that add more value to the business, thus being able to bring a number of improvements, but also some losses, as it provides an organizational reflection with many changes, not only in terms of development framework code. Before Scrum implementation there was no specific use of a methodology to support the management of the software process, which was an important factor in the search for a change. The adoption of Scrum was a necessary process to improve the company, arousing the interest of the team and improving organizational processes, which made the agile team and leading to better understanding of the results.

Keywords: Agile Methods. Software Engineering. Agile Culture. Scrum.



## LISTA DE FIGURAS

Figura 1 - Modelo Cascata.....	21
Figura 2 - Modelo Incremental .....	22
Figura 3 - Modelo Espiral.....	23
Figura 4 - Ciclo Scrum .....	30
Figura 5 - Cinco atividades de adaptação do Scrum .....	37
Figura 6 - Diagrama de atividades da abertura de solicitações antes da adoção do Scrum.....	40
Figura 7 - Diagrama de atividades da abertura de solicitações após a adoção do Scrum.....	44
Figura 8 - Ferramenta Asana .....	45
Figura 9 - Solicitações concluídas. ....	48
Figura 10 - Período médio para a conclusão das solicitações. ....	49
Figura 11 - Você considera que a mudança do processo de desenvolvimento no geral foi?.....	50
Figura 12 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior? .....	50
Figura 13 - Como você avalia a mudança de processo no contexto do gerenciamento de escopo (análise das novas requisições, divisão em tarefas, clareza nos requisitos, documentação)?.....	51
Figura 14 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior? .....	51
Figura 15 - Como você avalia a mudança de processo no contexto do gerenciamento de tempo (delegação de tarefas, estimativas de tempo, acompanhamento do desenvolvimento)?.....	52
Figura 16 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior?.....	52
Figura 17 - Como você avalia a mudança de processo no contexto do gerenciamento de qualidade (validação das implementações, gerenciamento dos releases)? .....	53
Figura 18 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior? .....	53
Figura 19 - Como você avalia a mudança de processo no contexto do gerenciamento de configuração (controle de versões, registro das mudanças, integração contínua)?.....	54
Figura 20 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior? .....	54
Figura 21 - Como você avalia o processo de Implantação do Scrum na empresa? .....	55
Figura 22 - Como você avalia a comunicação entre os membros da equipe após a implantação do Scrum?.....	55

## **LISTA DE TABELAS**

Tabela 1 - Solicitações antes da implantação do Scrum. ....	41
Tabela 2 - Papéis do Gerente de Projetos e Testes. ....	46
Tabela 3 - Solicitações depois da implantação do Scrum. ....	47

## **LISTA DE ABREVIATURAS E SIGLAS**

*ASD - Adaptive Software Development*

*DSDM - Dynamic Systems Development Method*

*FDD - Feature Driven Development*

*XP - Extreme Programming* – Programação Extrema

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
1.1 OBJETIVOS.....	16
1.1.1 Objetivo Geral .....	16
1.1.2 Objetivos Específicos.....	16
1.2 JUSTIFICATIVA .....	16
1.3 METODOLOGIA.....	17
1.4 ORGANIZAÇÃO .....	17
<b>2 PROCESSO DE SOFTWARE.....</b>	<b>19</b>
2.1 MODELOS .....	20
2.1.1 Modelo Cascata.....	20
2.1.2 Modelo Incremental .....	21
2.1.3 Modelo Espiral.....	22
2.2 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE .....	24
2.3 METODOLOGIA ÁGIL .....	25
2.3.1 Manifesto Ágil.....	25
2.4 PROGRAMAÇÃO EXTREMA .....	27
<b>3 SCRUM.....</b>	<b>29</b>
3.1 VISÃO GERAL.....	29
3.2 PILARES DO SCRUM .....	30
3.2.1 Transparência .....	30
3.2.2 Inspeção.....	31
3.2.3 Adaptação.....	31
3.3 PAPÉIS DO SCRUM .....	31
3.3.1 Scrum Master .....	31
3.3.2 Product Owner.....	32
3.3.3 Team .....	32
3.4 EVENTOS DO SCRUM .....	32
3.4.1 Sprint.....	32
3.4.2 Reuniões .....	33
3.5 ARTEFATOS DO SCRUM .....	35
3.6 ADAPTANDO-SE AO SCRUM.....	37
<b>4 A IMPLANTAÇÃO DO SCRUM .....</b>	<b>38</b>
4.1 O CENÁRIO DA EMPRESA .....	38

4.2 PROCESSO DE DESENVOLVIMENTO ANTES DA ADOÇÃO DO SCRUM .....	38
4.3 IMPLANTAÇÃO DO SCRUM.....	42
<b>4.3.1 Mudança de Papéis .....</b>	<b>46</b>
4.3 RESULTADOS .....	47
<b>4.3.1 Análise do questionário .....</b>	<b>49</b>
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>56</b>
<b>REFERÊNCIAS .....</b>	<b>57</b>



## 1 INTRODUÇÃO

As metodologias de desenvolvimento ágil, no Brasil, têm gerado grande entusiasmo entre seus usuários, assim como na comunidade acadêmica. Destacam-se, principalmente, os aspectos relacionados as melhorias nos resultados que a empresa de software deseja obter, como o aprimoramento de seus processos internos e de sua estrutura organizacional (VARASCHIM, 2009).

Nesse aspecto, destaca-se o Scrum, que vem ganhando cada vez mais espaço, pois tem como objetivo acelerar o desenvolvimento de software visando uma melhoria contínua do processo. Tendo como características: (i) a ênfase na comunicação em tempo real; (ii) a documentação reduzida comparando com outras metodologias; (iii) e, tratando também cada nova iteração no software como um novo projeto, que será realizado. Ressalta-se que essa metodologia visa agregar incrementos que estejam ligados diretamente ao valor do negócio (VARASCHIM, 2009).

Segundo Schwaber (2011), Scrum é um framework para desenvolver e manter produtos complexos, esta definição consiste em papéis, eventos, artefatos e as regras do Scrum que unem os demais e os mantêm integrados. Ken Schwaber e Jeff Sutherland desenvolveram o Scrum, o Guia do Scrum é escrito e fornecido por eles.

Confrontando essas características com os ambientes encontrados, hoje em dia, em empresas de desenvolvimento de software, verifica-se que essa metodologia se aplica muito bem, pois temos cenários aonde cada vez mais os desenvolvedores vêm trabalhando com pequenos incrementos e entregas mais curtas, como também cada vez mais se busca a qualidade dessas entregas, gerando maior segurança e valor para os clientes.

A implantação de uma metodologia ágil requer uma mudança não só em nível de ferramentas e organização da equipe, mas sim uma mudança real na mentalidade de cada participante do processo.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo Geral

Realizar a implantação do método ágil Scrum e um acompanhamento dos processos a fim de verificar as seguintes hipóteses:

1. A implantação de uma metodologia ágil reduz os prazos de entrega.
2. A implantação de uma metodologia ágil permite um melhor controle sobre as atividades desenvolvidas pela equipe.

### 1.1.2 Objetivos Específicos

- a) Estudar a metodologia ágil Scrum.
- b) Adaptar o Scrum para a realidade da empresa de desenvolvimento.
- c) Implantar o Scrum no setor de desenvolvimento.
- d) Analisar os impactos positivos e negativos que a aplicação da metodologia causou.

## 1.2 JUSTIFICATIVA

O desenvolvimento desse projeto será realizado em uma empresa de desenvolvimento de software. A mesma possui uma equipe de desenvolvimento formada por quatro participantes, que por sua vez desempenham papéis variados. Antes da implantação do Scrum não havia uso específico de uma metodologia para apoiar o gerenciamento do processo de software, que envolve desde a parte de análise de requisitos até a implantação do produto. Sendo assim, todas as atividades eram realizadas conforme fosse conveniente ao grupo naquele momento.

Neste contexto, as atividades dos projetos vinham sendo realizadas de forma não padronizada, e questões como, definição e distribuição de tarefas, definição e seleção de competências, definição da matriz de responsabilidades e canais de comunicação, eram até então tratadas com base na experiência e na disponibilidade dos participantes.

A falta de padronização para guiar a equipe algumas vezes causa problemas na troca de informações, o que é extremamente prejudicial,



visto que, a maioria das tarefas possui interdependência e essa conexão pode envolver diferentes tipos de *stakeholders*. Pelo fato de a equipe ser pequena, a comunicação não é algo tão complicado, mas com o aumento da equipe, a comunicação e a organização podem se tornar um problema. Assim, esses problemas podem implicar diretamente em aspectos negativos dentro da empresa, como a desmotivação de integrantes da equipe, por não ter uma forma padrão de se apresentar o que se tem a fazer, o que está sendo feito e o que já foi feito. Outra situação é a frequente alteração dos requisitos do projeto sendo necessária a simplicidade na busca dessas informações.

Outro fator relevante para a realização deste projeto de implantação de uma metodologia de desenvolvimento de software é a falta de documentação do software desenvolvido, que não acontece por falta de interesse por parte da equipe, mas sim pela falta de instrução, organização do tempo e conhecimento para desempenhar esse tipo de tarefa. Buscando solucionar esses problemas foi escolhida a metodologia ágil Scrum para ser adotada na empresa, pelo fato da equipe ser mostrar bastante favorável a esta adoção.

### 1.3 METODOLOGIA

Para atingir os objetivos propostos, foram realizadas as seguintes etapas:

- 1) A primeira etapa constituiu em uma revisão bibliográfica sobre as práticas, metodologias e *frameworks* ágeis, com foco no Scrum, visando identificar quais conceitos e práticas podem ser aplicadas;
- 2) A segunda etapa objetivou realizar uma pesquisa com a equipe de desenvolvimento da empresa, levantando os problemas existentes e apresentando as possíveis soluções com a utilização do Scrum;
- 3) Durante a terceira etapa foi feito um planejamento e implementação da metodologia;
- 4) E, a quarta e última etapa foram avaliados os impactos da mudança efetuada.

### 1.4 ORGANIZAÇÃO

O presente estudo está dividido em seis capítulos. O capítulo 1 diz respeito a introdução do que será apresentado no decorrer deste

trabalho de conclusão de curso, bem como seus objetivos gerais e específicos, justificativa e a metodologia empregada.

O capítulo 2 tem como objetivo apresentar a revisão bibliográfica das práticas que serão apresentadas, com foco em modelos de processo de software e nas metodologias ágeis desde o seu surgimento.

O capítulo 3 visa uma abordagem mais detalhada da metodologia de desenvolvimento ágil, Scrum. Detalhando seus objetivos e seus processos, afim de realizar a implantação da mesma.

O capítulo 4 apresenta o cenário da organização antes e depois em que o presente estudo foi realizado. Externando nas seções finais uma análise comparativa dos resultados obtidos.

O capítulo 5 foca na avaliação de todo o processo ao final da implantação da metodologia ágil e da avaliação dos resultados obtidos, apresenta também os benefícios obtidos pela empresa. Por fim, o capítulo 6 apresenta as oportunidades em aberto para trabalhos futuros.

## 2 PROCESSO DE SOFTWARE

Segundo Pressman (2010), a aflição que tomou conta das equipes de desenvolvimento de software no fim da década de 60 ficou conhecida como a Crise do Software. Na época foram encontrados problemas relacionados ao crescimento do mercado de desenvolvimento de softwares que apontaram as causas para a alta taxa de insucesso dos projetos de software que culminava com o estouro nos prazos, com a baixa qualidade dos produtos e a falta de documentação destes sistemas dificultando a manutenção dos mesmos. Como solução para estes problemas surgiu a engenharia de software com suas técnicas, ferramentas e métodos.

Segundo Hiram (2012), o termo “Engenharia de Software” surgiu no final da década de 60. Foi criado por Fritz Bauer em uma conferência patrocinada pelo Comitê de Ciência da Organização do Trato do Atlântico Norte, seguindo as mesmas definições da engenharia tradicional, com relação adequada ao custo/benefício do produto, que não falhe e que seja eficiente. Fritz Bauer também ressaltou que os sistemas devem formar uma estrutura matemática, construído em níveis e módulos. A base dessa estrutura deve ser o foco na qualidade, seguido por processos, métodos e ferramentas. Os métodos de desenvolvimento de software definem um canal de comunicação entre a equipe de desenvolvimento, utilizando padrões de trabalho a serem seguidos, baseados na maioria das vezes em duas abordagens: estruturada ou orientada a objetos.

Segundo Wirth (2012), a engenharia de software segue o conceito de disciplina na produção de software, fundamentado nas metodologias, que por sua vez seguem métodos que utilizam ferramentas automáticas para englobar as principais atividades do processo de produção.

Salienta-se que um processo de software é como uma receita a ser seguida para a execução de um projeto. Podemos definir também como uma sequência de passos a serem executados com a finalidade de alcançar algum objetivo (PAULA FILHO, 2009).

Muitos são os modelos de processo de software, cada um com suas características específicas, porém eles devem conter quatro atividades fundamentais: (i) a especificação de software, que trata da definição das funcionalidades e restrições do software; (ii) o projeto e a implementação de software, que garante que o software seja produzido para alcançar seus objetivos atendendo suas especificações; (iii) a validação do software, o mesmo deve passar por uma avaliação com o

intuito de garantir o atendimento às necessidades dos clientes; e, (iv) a evolução de software (SOMMERVILLE, 2011).

Os processos de software visam estabelecer, para a equipe de desenvolvimento, as atividades que devem ser realizadas para se obter um produto de software. Existem alguns modelos que são como representações desses processos, como, modelo cascata, modelo incremental e modelo espiral (SOMMERVILLE, 2011).

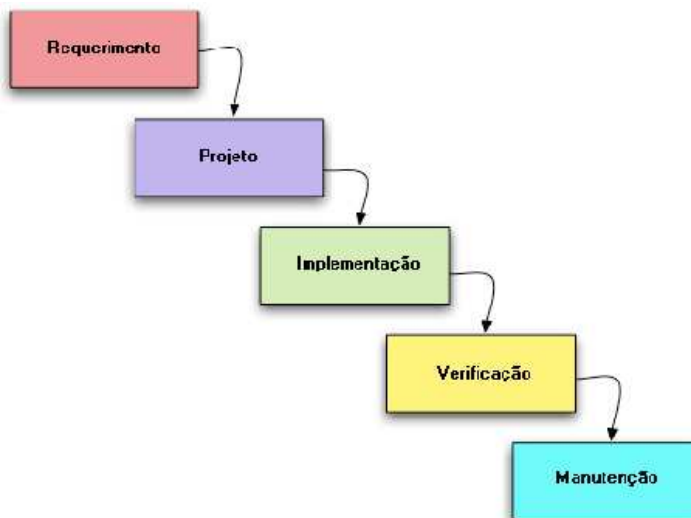
## 2.1 MODELOS

### 2.1.1 Modelo Cascata

Derivando dos processos mais tradicionais da engenharia, o modelo clássico ou cascata, que também é conhecido por abordagem “*top-down*”, foi proposto na década de 70, com o objetivo de estabelecer a ordem no desenvolvimento de grandes produtos de software. Até meados da década de 1980, era praticamente o único modelo utilizado. Neste, o desenvolvimento deve ser dirigido a planos, ou seja, deve haver um planejamento de todas as atividades do processo (SOMMERVILLE, 2011).

O modelo cascata é um dos mais importantes, e é referência para muitos outros, pois serve de base para muitos projetos modernos. A versão original deste modelo foi melhorada ao longo do tempo e continua sendo muito utilizada hoje em dia (RAMOS et al., 2014).

Segundo Hirama (2012), podemos destacar algumas características desse processo, tais como: seguir uma disciplina nas tarefas de especificação, codificação e testes, nunca se inicia uma atividade sem que a atividade anterior tenha sido terminada, a sequência de atividades é rígida e por fim o usuário/cliente só participa do início e fim do projeto.



**Figura 1 - Modelo Cascata**  
**Fonte: (RAMOS et al., 2014).**

Segundo Jubileu (2008), apesar do modelo cascata ser muito conhecido e utilizado, ele pode apresentar alguns problemas, dentre estes: projetos reais dificilmente irão seguir um fluxo sequencial como proposto pelo modelo; no início do projeto é muito difícil estabelecer todos os requisitos e o produto demora a ser entregue para o cliente.

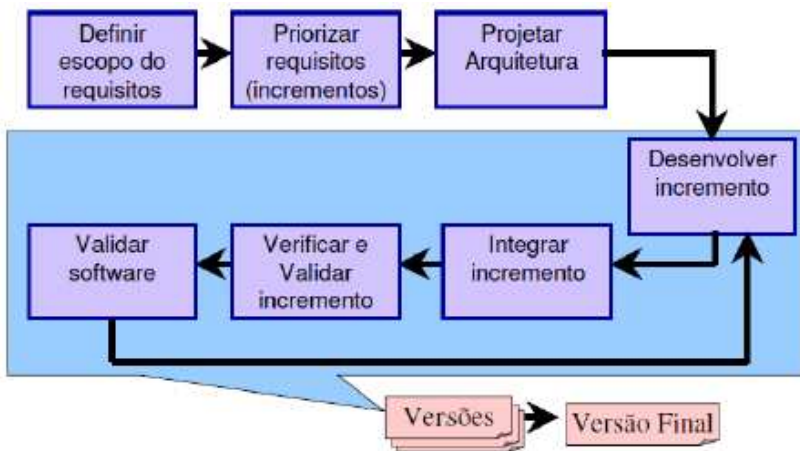
Processos rígidos como os do modelo cascata tendem a ser utilizados somente no desenvolvimento de sistemas de níveis críticos em segurança e/ou de proteção, que exigem conhecimentos especializados. Para a maioria dos sistemas desenvolvidos, atualmente, esse modelo não apresenta grandes benefícios perante outras abordagens (SOMMERVILLE, 2011).

### **2.1.2 Modelo Incremental**

Esse modelo segue uma ideia parecida com o modelo em cascata, porém mais iterativa. Tem como objetivo reduzir o retrabalho no processo de desenvolvimento, proporcionando ao cliente adiar algumas decisões mais detalhadas sobre alguns requisitos, identificando inicialmente as tarefas mais prioritárias que o sistema deve obter, e

posteriormente definindo uma série de entregas com novas funcionalidades de acordo com a expectativa do cliente (JUBILEU, 2008).

Segundo Sommerville (2011), esse modelo é mais eficaz do que a abordagem em cascata citado na seção anterior. Para a maioria dos sistemas de negócios, como *e-commerce* e sistemas pessoais, raramente se identificam todos os requisitos dos sistemas inicialmente, pois é normal que a cada novo incremento realizado haja mais conhecimento adquirido em relação ao objetivo final.



**Figura 2 - Modelo Incremental**

**Fonte:** (JUBILEU, 2008).

Segundo Jubileu (2008), as principais vantagens deste modelo são: (i) não há necessidade de esperar uma versão final para iniciar o uso do software; (ii) as primeiras entregas podem ser usadas como um tipo de protótipo, visando definir melhor requisitos e incrementos futuros; (iii) e menor risco de fracasso do software, uma vez que entregando inicialmente as funções mais prioritárias se torna inevitável que as mesmas passem por um período de testes mais intensos.

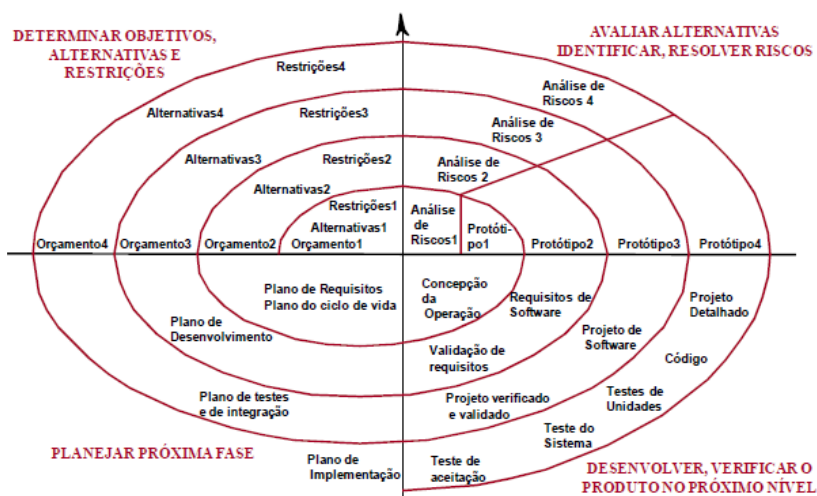
### 2.1.3 Modelo Espiral

Segundo Paula Filho (2009), este modelo é diferente dos modelos citados anteriormente, principalmente do modelo em cascata. Usa-se uma abordagem cíclica para desenvolver um sistema, resultando

em entregas incrementais, combinando a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos, que requerem uma equipe de desenvolvimento experiente e disciplinada.

O ciclo de vida no formato de uma espiral permite adicionar ao modelo novos requisitos durante o fluxo. Apresenta alta flexibilidade e maior visibilidade, dando a oportunidade ao cliente de avaliar as partes entregues do produto e assim fornecer melhores dados sobre os requisitos ou até alterá-los. Porém, toda essa flexibilidade requer uma gestão sofisticada do projeto para que sua estrutura continue sólida a cada iteração entregue (PAULA FILHO, 2009).

A Figura 3 é um esboço do funcionamento do modelo:



**Figura 3 - Modelo Espiral**

**Fonte: (JUBILEU, 2008).**

Segundo Jubileu (2008), cada ciclo mostrado na Figura 3 envolvem quatro atividades principais: (i) a elaboração dos objetivos de processo e produto, (ii) avaliação das alternativas em relação aos objetivos, identificando e solucionando os riscos encontrados, (iii) definição do produto e dos processos a serem realizados, (iv) planejamento do próximo ciclo, incluindo partições do sistema para serem distribuídos em novos ciclos paralelos.

Este modelo possui algumas variantes, como o modelo de prototipagem evolutiva, onde a espiral não é utilizada para o

desenvolvimento de todo o produto, mais sim de uma série de versões provisórias com protótipos, com finalidade de chegar a um produto final. Outra variante seria o modelo de entrega evolutiva, no qual as definições referentes à estrutura do produto são feitas em cascata e os passos seguintes em espiral (PAULA FILHO, 2009).

## 2.2 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Já faz alguns anos que o desenvolvimento de software deixou de ser sinônimo apenas de código. Hoje em dia se faz necessário a utilização de uma metodologia para a realização dos inúmeros processos que envolvem esse tipo de trabalho.

Segundo Houaiss (2001), metodologia é um conjunto de métodos, princípios e regras que estabelecem normas para a realização de uma atividade. Métodos seriam os caminhos para se atingir uma meta, como um modo de agir, uma técnica ou uma forma de organização.

Para facilitar o entendimento, podemos citar alguns objetivos de uma metodologia no âmbito do desenvolvimento de software. Segundo Wirth (2012), são eles: definir de forma clara “quem” faz “o que”, “quando”, “como”, e até mesmo “onde”, para todos os que estejam envolvidos diretamente ou não com o desenvolvimento de software, definindo também qual o papel dos técnicos, dos usuários, e o da administração da empresa no processo de desenvolvimento.

Com estes objetivos bem definidos evita-se a situação em que o conhecimento sobre o sistema é de poucos. Além disso, deve-se instruir um conjunto de padrões preestabelecidos, de modo a se evitar a subjetividade na abordagem e a fim de garantir fácil integração entre os sistemas desenvolvidos. Com isso, o uso de uma metodologia possibilita (LEITE, 2014):

- Ao gerente: controlar o projeto de desenvolvimento de software, mantendo foco no objetivo do projeto com controle para que não haja desvios de planejamento de custos e prazos, que, se negligenciados ou mal conduzidos, podem pôr em risco o sucesso do projeto.
- Ao desenvolvedor: obter a base para produzir de maneira eficiente um software de qualidade que satisfaça os requisitos estabelecidos.

Segundo Leite (2014) é importante ressaltar que uma metodologia não deve limitar a criatividade profissional, mas sim servir



como instrumento de planejamento, com o intuito de permitir fácil entendimento, comunicação e coordenação das partes envolvidas.

A escolha de uma metodologia a ser utilizada no desenvolvimento é de suma importância e deve ser realizada com base na natureza do projeto e do produto a ser desenvolvido, dos métodos e ferramentas a serem utilizadas e dos controles e produtos intermediários desejados (WIRTH, 2012).

## 2.3 METODOLOGIA ÁGIL

### 2.3.1 Manifesto Ágil

Devido ao pouco sucesso obtido nos projetos de desenvolvimento de software, diversas iniciativas surgiram visando encontrar maneiras mais eficientes de desenvolver sistemas complexos. Esses estudos deram origem a algumas metodologias com foco mais nos fatores humanos e na satisfação do cliente do que na burocracia dos processos (SILVA, 2013).

Santos (2013) cita que em 2001, especialistas se reuniram para discutir e tentar estabelecer uma nova metodologia de desenvolvimento, porém o grupo não conseguiu estabelecer uma metodologia comum, levando em consideração a complexidade de todos os processos de desenvolvimento de software. Obtiveram como resultado dessa reunião o manifesto ágil que se baseia em valores como:

1. Indivíduos e interações são mais importantes do que processos e ferramentas;
2. Softwares em funcionamento são mais importantes do que documentação abrangente, que por muitas vezes consome muito tempo;
3. Foco no cliente, pois considera que a colaboração com o mesmo é mais importante do que a negociação de contratos;
4. Adaptação a mudanças é mais importante do que seguir o plano inicial, tendo em vista as mudanças que ocorrem durante o processo de desenvolvimento.

Tendo esses valores como base foram elaborados os princípios das metodologias ágeis (SANTOS, 2013):

1. Uma das maiores prioridades é pela satisfação do consumidor por meio de entregas contínuas de valor e em um curto período de tempo;

2. Mudanças de requisitos são bem vindas mesmo em estágios avançados do desenvolvimento. Processos ágeis aproveitam as mudanças em benefícios da vantagem competitiva do cliente;
3. Entregar o produto funcionando em curto período;
4. Desenvolvedores e gestores devem trabalhar diariamente em conjunto;
5. Criar projetos com as pessoas motivadas. Confie nelas e de suporte e ambiente para que o trabalho seja feito;
6. O método mais eficiente e eficaz de transmitir informações em um projeto é sempre informando diretamente ao cliente;
7. Produto funcionando é a principal medida para o progresso;
8. Processos ágeis promovem o desenvolvimento sustentável. As empresas contratantes, os desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
9. Atenção contínua excelência técnica e ao design melhoram a agilidade;
10. Simplicidade. A arte de deixar de fazer trabalhos desnecessários é essencial;
11. Os melhores requisitos, arquiteturas e design surgem de equipes que praticam a autogestão;
12. Em um determinado período de tempo a equipe deve repensar sobre como se tornar mais eficaz. Após a reflexão deve reajustar-se de acordo com as necessidades percebidas.

Segundo Cohn (2011), muitas empresas estão tentando tornar suas equipes de desenvolvimento mais ágeis, pois as mesmas vêm produzindo software com mais qualidade e que atendem melhor os requisitos impostos pelos clientes. Equipes ágeis tendem a introduzir seus produtos no mercado com muito mais rapidez, levando em consideração principalmente a satisfação dos clientes com os requisitos e objetivos do produto.

Ao longo dos anos surgiram diversas metodologias de desenvolvimento ágil, como por exemplo, Scrum, Crystal Clear, Adaptive Software Development (ASD), Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), e principalmente Extreme Programming (XP), o mais difundido de todos

(PAULA FILHO, 2009) que será apresentado com mais detalhes nesse estudo juntamente como o Scrum.

## 2.4 PROGRAMAÇÃO EXTREMA

Geralmente chamado de XP, o Extreme Programming é talvez um dos métodos ágeis mais conhecido e utilizado. Tem como base o desenvolvimento iterativo, com grande participação do cliente, que define um grupo de funções chamadas de história de usuário (PAULA FILHO, 2009).

Segundo Hirama (2012), o XP apresenta uma série de características que refletem os princípios dos métodos ágeis, são elas:

- a) Planejamento incremental – o desenvolvimento é dividido em liberações, cada liberação possui um conjunto de iterações;
- b) Pequenos *releases* – Desenvolvimento de pequenas versões de software, fornecendo valor ao negócio;
- c) Projeto simples – Cada projeto deve atender somente as necessidades atuais;
- d) Desenvolvimento de testes antes do código – é mais fácil implementar código após conhecer os testes que serão impostos;
- e) Refatoração do código – os desenvolvedores devem modificar o código sempre que possível, visando facilitar sua manutenção futura;
- f) Programação em pares – os desenvolvedores trabalham em duplas em um mesmo computador, melhorando a qualidade do código e diminuindo erros;
- g) Propriedade coletiva – todos os desenvolvedores trabalham em todas as áreas do sistema, podendo alterar qualquer parte do código, desde que atenda os testes;
- h) Integração contínua – após a conclusão de uma tarefa, todo o código é reintegrado em um repositório, sendo novamente testado;
- i) Ritmo sustentável de trabalho – para melhor qualidade de código, os desenvolvedores devem estar motivados e completarem a iteração planejada;
- j) Cliente no local em tempo integral – um representante do cliente fica disponível no local, como parte da equipe.

O XP não é recomendado para desenvolver softwares complexos e de grande porte ou críticos, sendo utilizados atualmente para pequenos projetos e tipicamente para ambientes Web (HIRAMA, 2012).

### 3 SCRUM

Segundo Scrum (2014), o Scrum é uma metodologia ágil de desenvolvimento de software. Como visto anteriormente, é utilizado para definir padrões de gestão e planejamento de projetos de software. Os processos são divididos em ciclos, sendo que cada um representa uma parte na qual o conjunto de atividades deve ser executado. Neste capítulo vamos conhecer melhor o Scrum.

#### 3.1 VISÃO GERAL

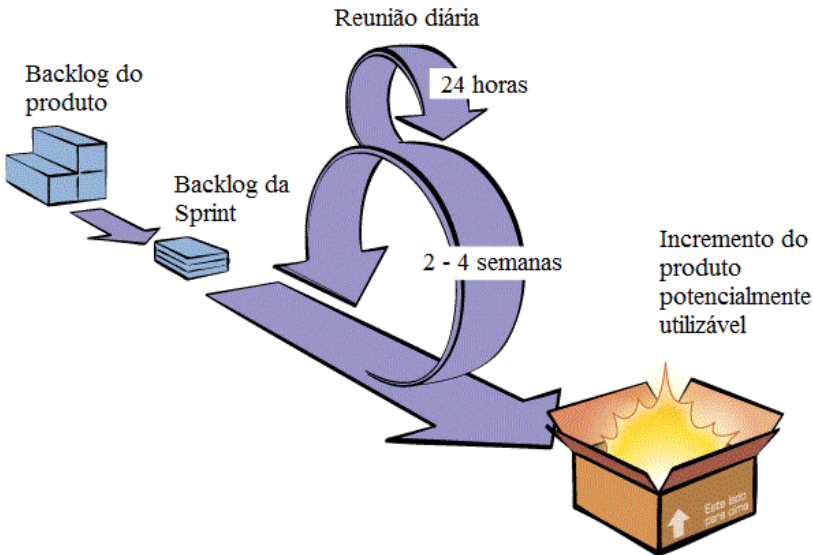
O Scrum utiliza um esqueleto de processo iterativo e incremental para aperfeiçoar a previsibilidade e melhorar o controle de riscos. As equipes são enxutas e possuem três papéis principais desempenhados no projeto: o *Product Owner*, o *Scrum Master* e o time de desenvolvimento (SCHWABER, 2004):

- *Product Owner* (Dono do Produto): representa os interesses do cliente no projeto e, em alguns casos, é o próprio cliente;
- *Scrum Master* (Mestre Scrum): responsável pela execução de todas as regras do Scrum;
- *Team* (Time): responsável por desenvolver o projeto.

Segundo Albino, Souza e Prado (2013), o projeto Scrum inicia quando uma visão do que deverá ser feito é criada, ou seja, se tem conhecimento das características que o cliente espera que o projeto contenha ao seu final, levando em consideração o problema atual. Após o entendimento do que será feito é criado um documento contendo a lista de todos os requisitos que foram levantados, conhecido como o *Product Backlog*.

Schwaber (2004), explica que após o lista de requisitos ser formada inicia-se uma reunião de planejamento chamada *Sprint Planning Meeting*, visando definir o *Sprint* inicial do projeto. Neste o *Product Owner* e o *Team* decidem em conjunto o que deverá ser desenvolvido. Ao longo do *Sprint* reuniões são feitas diariamente para acompanhar o progresso do trabalho e outras reuniões podem ser agendadas, se necessário. Ao final do *Sprint*, uma *Sprint Review Meeting* (reunião de revisão) é realizada para que seja apresentado o resultado alcançado. Neste instante, são validadas as funcionalidades e caso sejam necessárias, adaptações são realizadas. Esse processo se repete até que todo o *Product Backlog* seja atendido e o produto

entregue ao cliente como desejado. A Figura 4 apresenta uma ilustração do ciclo do Scrum:



**Figura 4 - Ciclo Scrum**  
Adaptado de (SCRUM, 2014).

### 3.2 PILARES DO SCRUM

O Scrum é fundamentado nas teorias empíricas de controle de processo, que afirmam que o conhecimento vem da experiência e da tomada de decisões baseadas no que é conhecido. Segue assim três pilares: a transparência, a inspeção e a adaptação (SCHWABER; SUTHERLAND, 2013).

#### 3.2.1 Transparência

A transparência garante que os aspectos significativos do processo estejam visíveis aos responsáveis pelos resultados, requerendo aspectos definidos por um padrão comum para que todos tenham um mesmo entendimento do que está sendo apresentado. Por exemplo, uma definição de “pronto” deve ser obtido tanto pelos responsáveis pela criação quanto pelos responsáveis pela validação do produto (SCHWABER; SUTHERLAND, 2013).

### 3.2.2 Inspeção

A inspeção determina que, frequentemente, deve-se inspecionar os artefatos gerados bem como o progresso do projeto em direção ao objetivo, para detectar indesejáveis variações, porém, sem ter uma frequência que atrapalhe a própria execução das tarefas. As inspeções são mais benéficas quando realizadas de forma diligente por inspetores especializados no trabalho a se verificar (SCHWABER; SUTHERLAND, 2013).

### 3.2.3 Adaptação

A adaptação conclui que, quando determinado por uma inspeção que um ou mais aspectos de um processo estão fora dos limites aceitáveis, gerando um produto onde o resultado não será o esperado, ajustes devem ser realizados com urgência, minimizando os riscos e melhores resultados (SCHWABER; SUTHERLAND, 2013).

A inspeção e adaptação podem ser feitas de quatro formas:

- Reunião de planejamento da *Sprint*
- Reunião diária (*Daily Scrum Meeting*)
- Reunião de revisão da *Sprint*
- Retrospectiva da *Sprint*

## 3.3 PAPÉIS DO SCRUM

Segundo Silva (2013), o modelo de equipe é projetado para aperfeiçoar a flexibilidade, a criatividade e a produtividade. Contém três papéis principais, que são responsáveis pela divisão das responsabilidades, são eles: o *Scrum Master*, o *Product Owner*, e o *Team*.

### 3.3.1 Scrum Master

O *Scrum Master* é responsável pelo processo Scrum, por garantir que seja entendido e aplicado, por implementá-lo - fazendo dele uma cultura na organização - e ainda por garantir que toda a equipe siga as regras e as práticas do Scrum (SCHWABER; SUTHERLAND, 2013).

É responsável também por proteger a equipe e assegurar que ela não se comprometa excessivamente com relação àquilo que é capaz de

realizar durante um *Sprint*. Atua como intermediador na *Daily Scrum Meeting* e torna-se responsável por remover quaisquer obstáculos que sejam levantados pela equipe durante essas reuniões. Pode ser exercido por qualquer membro da equipe, porém, normalmente é exercido por um gerente de projetos (SCHWABER; SUTHERLAND, 2013).

### 3.3.2 Product Owner

O *Product Owner* ou dono do produto é o responsável por maximizar o valor do produto e por representar os interesses dos *stakeholders* no projeto. O *Product Owner* é a pessoa que irá gerenciar todos os itens do *backlog* do produto, garantido que o mesmo seja visível, transparente e claro para todos (SCHWABER; SUTHERLAND, 2013).

Toda a organização deve respeitar as suas decisões, uma vez que qualquer alteração no *backlog* deve ser feita pelo *Product Owner* (SCHWABER; SUTHERLAND, 2013).

### 3.3.3 Team

O *Team Scrum* é a equipe de desenvolvimento, que possui normalmente entre 6 a 10 pessoas. As principais características são: auto organizáveis, auto gerenciáveis e multifuncionais. Devem assim ser estruturadas e autorizadas para se auto organizarem e gerenciarem seu próprio trabalho (SCHWABER; SUTHERLAND, 2013).

## 3.4 EVENTOS DO SCRUM

Eventos são criados para criar uma rotina, com o intuito de diminuir reuniões não definidas, em que todo evento tenha uma duração máxima. Garante-se assim que não haja perdas no processo de planejamento, permite-se transparência e uma inspeção criteriosa e gera-se a oportunidade de adaptação de alguma coisa (SCHWABER; SUTHERLAND, 2013).

### 3.4.1 Sprint

O *Sprint* é o principal evento do Scrum. É o período onde incrementos “prontos” são criados e que normalmente possuem durações entre um mês. Uma nova *Sprint* se inicia após a conclusão da *Sprint* anterior (SCHWABER; SUTHERLAND, 2013).



Cada *Sprint* possui uma definição do que deve ser construído. Durante o processo o escopo é detalhado e revisado entre o time e o *product owner*, podendo ser clarificado e renegociado, desde que não afete os objetivos inicialmente descritos. Além disso, não é recomendado alterar a composição do time ou a sua duração (SCHWABER; SUTHERLAND, 2013).

Algumas situações podem gerar o cancelamento do *Sprint*, como por exemplo, quando os objetivos iniciais se tornam obsoletos. Porém, em função de sua curta duração é pouco provável que isto aconteça, e caso ocorra, esta tarefa cabe apenas ao *product owner*, mesmo que por recomendação das partes interessadas, do time de desenvolvimento ou do *Scrum master*. Após o cancelamento, os itens do *backlog* do *Sprint* que já haviam sido desenvolvidos podem ser entregues caso o *product owner* os aprove (SCHWABER; SUTHERLAND, 2013).

### 3.4.2 Reuniões

Existem nessa metodologia quatro tipos de reuniões:

1. Reunião de planejamento, é a reunião onde serão definidos os itens a serem desenvolvidos durante a *Sprint*, de forma colaborativa por toda a equipe de desenvolvimento. A mesma é dividida em duas etapas e tem uma duração média de oito horas para um *Sprint* de um mês, a qual deve ser proporcional ao tempo do *Sprint* (SCHWABER; SUTHERLAND, 2013).

Na primeira etapa, o *product owner* é o responsável por gerenciar o *backlog* do produto, mantendo os itens ordenados por prioridade, e o apresentando para o time de desenvolvimento, assim toda a equipe colabora na compreensão do trabalho que deve ser realizado. O único trabalho da equipe de desenvolvimento nesta etapa é o de selecionar o número de itens que podem ser desenvolvidos, levando em consideração, o desempenho obtido na *Sprint* anterior. Por fim, o time de desenvolvimento define o objetivo do *Sprint*, que irá orientá-los durante o desenvolvimento informando o motivo de o incremento estar sendo criado (SCHWABER; SUTHERLAND, 2013).

Na segunda etapa, o time de desenvolvimento decide como os itens selecionados serão desenvolvidos e detalha as estimativas de cada item selecionado, gerando o *backlog* do *Sprint*. É muito importante a presença do *product owner* para

ajudar na compreensão e na priorização dos itens e das decisões importantes. Se o time de desenvolvimento verificar que foram selecionados poucos ou muitos itens a serem desenvolvidos, eles serão analisados novamente com o *product owner*, podendo ser renegociados. Nesta fase, pessoas de outras equipes podem ser convidadas para fornecer opinião técnica ou de domínios específicos. Ao final da reunião de planejamento, o time de desenvolvimento deve ser capaz de explicar ao *product owner* e ao *Scrum master* como pretendem trabalhar para atingir os objetivos descritos (SCHWABER; SUTHERLAND, 2013).

2. Reunião diária, é um evento com 15 minutos de duração, com objetivo de que a equipe de desenvolvimento possa analisar o progresso do dia anterior. Resultando no planejamento do trabalho a ser realizado nas próximas 24 horas, devendo ser realizada sempre no mesmo local todos os dias, reduzindo a complexidade. Durante a reunião, cada membro do time de desenvolvimento deve responder às seguintes (SCHWABER; SUTHERLAND, 2013):

- O que foi feito desde a última reunião?
- O que será feito até a próxima reunião?
- O que está impedindo na conclusão do trabalho?

Com estas respostas avalia-se o progresso em direção ao objetivo final. Pode-se também identificar e remover qualquer problema que possa impactar na conclusão das atividades do *backlog* ao final do *Sprint* (SCHWABER; SUTHERLAND, 2013).

Reuniões diárias melhoram as comunicações, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, destacam e promovem rápidas tomadas de decisão, e melhoram o nível de conhecimento da Equipe de Desenvolvimento. Esta é uma reunião chave para inspeção e adaptação (SCHWABER; SUTHERLAND, 2013).

3. A reunião de revisão do *Sprint*, ocorre somente no fim do mesmo. Conta com a participação de toda a equipe do projeto e das partes interessadas, visando inspecionar o incremento gerado e adaptar o *backlog* do produto, se for necessário. É

recomendada uma duração de quatro horas para um *Sprint* de um mês, sendo que esse tempo deve ser proporcional ao tamanho do *Sprint* (SCHWABER; SUTHERLAND, 2013).

A reunião deve incluir as seguintes etapas: o *product owner* identifica quais itens foram realizados e quais não foram; em seguida, o time de desenvolvimento aponta os pontos positivos e negativos, informa como os problemas foram resolvidos e apresenta os itens que foram concluídos. Após esse resumo do *Sprint* passado, todos os participantes colaboram para definir quais serão os próximos passos, fornecendo dados para a reunião de revisão do próximo *Sprint* (SCHWABER; SUTHERLAND, 2013).

O resultado desta reunião é um *backlog* do produto revisado por todos os envolvidos no projeto, definindo uma lista de itens que serão realizados no próximo *Sprint*, podendo ser alterados ou não. Se novas oportunidades surgirem, o *backlog* do produto pode ser ajustado completamente (SCHWABER; SUTHERLAND, 2013).

4. A reunião de retrospectiva do *Sprint* tem foco no processo, sendo uma oportunidade para toda a equipe Scrum se inspecionar, visando à criação de um plano de melhorias para os próximos trabalhos, e dando ênfase na análise das pessoas, das relações, dos processos e das ferramentas. Ressalta-se que deve haver um período de três horas para um *Sprint* de um mês, o qual deve ser proporcional ao tamanho do *Sprint* (SCHWABER; SUTHERLAND, 2013).

O Scrum *master* deve incentivar a equipe a buscar melhorias no trabalho baseando-se nos pilares do Scrum e permitindo que a equipe de desenvolvimento se auto gerencie (SCHWABER; SUTHERLAND, 2013).

### 3.5 ARTEFATOS DO SCRUM

Os artefatos definidos para o Scrum têm como objetivo maximizar a transparência das informações para a equipe, assegurando assim maior sucesso na entrega de cada incremento pronto (SCHWABER; SUTHERLAND, 2013).

Os principais artefatos são: o *backlog* do produto, o *backlog* do *Sprint* e o incremento do produto (SCHWABER; SUTHERLAND, 2013).

- 1) O *backlog* do produto é uma lista de funcionalidades ou características que são necessárias no produto, sendo esses itens ordenados por prioridade. A gerência dessa lista é feita pelo *product owner*, sendo ele o responsável por incluir conteúdo, disponibilizar e ordenar todos os itens. Um *backlog* do produto nunca está completo, evoluindo como produto em consequência de um melhor entendimento de seus requisitos (SCHWABER; SUTHERLAND, 2013).

Esta lista é formada por características, funções, requisitos, melhorias e correções, possuindo para cada item, descrição do incremento, ordem e estimativa de tempo para produção. A ordenação geralmente é feita por valor, riscos, prioridades ou necessidades dos clientes (SCHWABER; SUTHERLAND, 2013).

Os itens que estão no topo da lista são os que possuem maior prioridade. Devem assim possuir um maior nível de detalhamento para serem encaminhados para o desenvolvimento, incluindo o esforço estimado para desenvolvê-lo, que é uma tarefa de responsabilidade da equipe de desenvolvimento. Quando encaminhados passam para uma fase onde se encontram disponíveis ou então executáveis, podendo assim serem selecionados para o próximo *Sprint* (SCHWABER; SUTHERLAND, 2013).

- 2) O *backlog* do *Sprint* é uma lista que contém os itens que foram selecionados do *backlog* do produto para o desenvolvimento, juntamente com o plano de entrega e o objetivo da *Sprint*, o seja, é o planejamento do que será entregue no próximo incremento do produto (SCHWABER; SUTHERLAND, 2013).

O *backlog* do *Sprint* tem como objetivo tornar visível todo o trabalho que o time de desenvolvimento deve realizar, podendo estar em constante alteração durante a *Sprint*, levando em consideração que a cada tarefa que é realizada novos aprendizados ou entendimentos vão surgindo (SCHWABER; SUTHERLAND, 2013).

O monitoramento do trabalho é feito através da reunião diária, que permite um entendimento geral sobre o andamento da *Sprint*, estimando o tempo restante com

mais precisão e facilitando a identificação de possíveis atrasos no desenvolvimento. Caso um atraso seja identificado, na próxima reunião diária este desvio pode ser corrigido, reduzindo o impacto na entrega final (SCHWABER; SUTHERLAND, 2013).

- 3) O incremento do produto é o conjunto de itens que foram desenvolvidos durante a *Sprint*, devendo estar em condições de uso, independentemente da decisão do *product owner* por liberá-los ou não. Os resultados desse incremento devem ser avaliados para auxiliar no planejamento das próximas *Sprints*, estimando melhor o tempo necessário e quantidade de itens que podem ser selecionados (SCHWABER; SUTHERLAND, 2013).

### 3.6 ADAPTANDO-SE AO SCRUM

Segundo Cohn (2011), cinco atividades comuns definidas por Lori Schubring - gerente de desenvolvimento de uma grande empresa e umas das primeiras a perceber que uma mudança era necessária no processo de desenvolvimento utilizado - são base para uma adoção bem sucedida e duradoura do Scrum. Para iniciar a mudança é fundamental que seja obtido um reconhecimento de que o processo atual não está dando os resultados esperados. Se faz necessário também o desejo da adoção do Scrum como ferramenta de solução. Para isso, possuir aptidão, habilidade e um ambiente onde haja uma promoção do Scrum são fundamentais por haver compartilhamento de experiências, que irão tornar possível o conhecimento de todos sobre os benefícios da adoção, refletindo na transferência desses benefícios para toda a organização.



**Figura 5 - Cinco atividades de adaptação do Scrum**  
**Fonte: (COHN, 2011).**

## 4 A IMPLANTAÇÃO DO SCRUM

Este capítulo apresenta o cenário encontrado na empresa onde o Scrum foi implantado, demonstrando os cenários do antes e do depois da adoção do Scrum. Como parte do processo de avaliação do Scrum na Empresa foi aplicado um questionário com o objetivo de verificar o nível de satisfação da equipe após a mudança na cultura organizacional.

Com o objetivo de preservar a empresa e seus objetivos, não será divulgado o seu nome. Para melhor entendimento será tratada com o nome fictício “Empresa”.

### 4.1 O CENÁRIO DA EMPRESA

A Empresa possui uma estrutura pequena em relação ao espaço físico e ao quadro de funcionários. É jovem, com apenas cinco anos de idade, mas com profissionais com mais de dez anos de experiência no mercado. Atualmente atende nove clientes, desenvolvendo um sistema específico que é utilizado por mais ou menos cinco mil usuários, sendo que desse montante uma média de cem usuários fazem solicitações, o que demanda uma organização bem apurada quando se trata de novas iterações no produto, levando em conta também que esse produto é dividido em cinco módulos independentes.

No cenário atual, a Empresa conta com quatro colaboradores no setor de desenvolvimento, sendo que cada um possui um papel bem definido, sendo eles: um analista de sistemas, um programador, um gerente de projetos e testes, e um profissional de testes.

A preocupação com os testes é bastante grande, pois a empresa presa pela qualidade do software desenvolvido, e também pelo mínimo de transtorno para o cliente na implantação de novas funcionalidades.

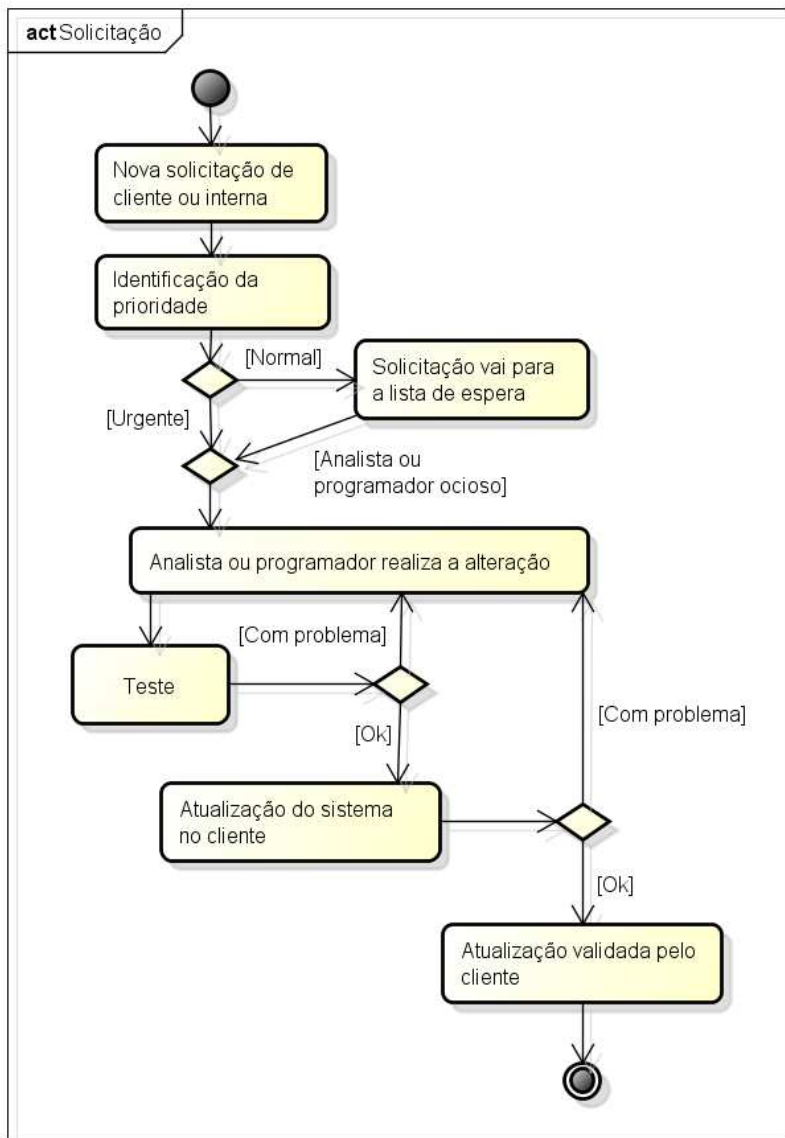
Destaca-se também o conhecimento de todos os participantes da equipe com relação ao segmento em que o produto se encaixa, tendo em vista que para que sejam efetuadas entregas ditas como de qualidade, devem atender as solicitações propostas.

### 4.2 PROCESSO DE DESENVOLVIMENTO ANTES DA ADOÇÃO DO SCRUM

Um dos problemas que resultaram no interesse pela adoção de uma metodologia de desenvolvimento se deve principalmente ao fato de que a empresa não utilizava nenhum processo explícito de

desenvolvimento de software. O fluxo do processo de desenvolvimento pode ser descrito da seguinte forma, conforme a Figura 6:

1. Uma nova solicitação é feita por algum *stakeholder*.
2. Quando considerada “urgente” o próprio analista resolve ou passa para o programador. Se considerada normal, vai para uma lista de espera para ser desenvolvida; esta lista é separada por clientes ou por módulos, mas sem ordenação de prioridades.
3. Com o decorrer do trabalho, programador e analista realizam esses incrementos, sem prazo de entrega definido.
4. Após alguns incrementos é gerada uma nova atualização que será testada e posteriormente entregue aos clientes.



**Figura 6 - Diagrama de atividades da abertura de solicitações antes da adoção do Scrum.**



O maior problema deste processo é a falta de controle sobre essas solicitações, gerando problemas de comunicação entre todos os envolvidos no processo, incluindo os solicitantes. Além disso, ao aparecer uma solicitação definida como “urgente” eram necessárias muitas horas extras, levando a compreensão por parte da equipe de que o tempo estava sendo mal empregado, ferindo o conceito de qualidade, que a Empresa busca em seu produto e em seus processos.

Todo o controle das atividades e solicitações, assim como das entregas, era feito por uma ferramenta desenvolvida pela própria equipe e em tabelas.

As solicitações eram divididas em seções, sendo essas nomeadas por clientes ou módulos do produto. Os itens eram descritos sem um nível de detalhamento adequado, sendo atribuído um responsável pela atividade inserida. Este deveria dar mais detalhes sobre o incremento quando fosse ser desenvolvido.

Dois problemas ocorriam frequentemente: no primeiro era normal encontrar solicitações iguais presentes em mais de uma seção, como por exemplo, na seção que indicava algum cliente e o módulo; o segundo era o baixo nível de detalhamento das solicitações, em que o responsável por indicá-las já não tinha em mãos os dados necessários para realizar o incremento. Desta forma, havia transtorno não só dentro da equipe, mas também ao solicitante, pois era necessário que o mesmo especificasse novamente o que havia solicitado.

A Tabela 1 mostra os dados referentes a solicitações disponíveis no sistema utilizado na empresa, com dados de até 8 meses anteriores a adoção do Scrum.

**Tabela 1 - Solicitações antes da implantação do Scrum.**

<b>Mês / Ano</b>	<b>Solicitações cadastradas</b>	<b>Solicitações concluídas</b>	<b>Período médio para conclusão</b>
Dez / 2013	4	0	-
Jan / 2014	27	7	33 dias
Fev / 2014	6	0	-
Mar / 2014	3	4	124 dias
Abr / 2014	16	0	-
Mai / 2014	18	1	44 dias
Jun / 2014	13	11	127 dias
Jul / 2014	43	37	34 dias
<b>Totais</b>	<b>130</b>	<b>60</b>	<b>72 dias</b>
<b>Média mensal</b>	<b>16</b>	<b>7,5</b>	<b>-</b>

Uma observação feita com base nesta tabela chama a atenção para o fato de os meses de Fevereiro e Abril não terem solicitações concluídas. Ao analisar com a equipe verificou-se que nesses dois meses estava sendo implantado o sistema em dois novos clientes, voltando os esforços da equipe para este ponto. O período médio de dias para a conclusão das solicitações também está elevado, sendo que foi encontrado o dobro de solicitações novas em relação às concluídas.

No mês de julho há mais solicitações cadastradas devido ao fato de a equipe tentar reduzir o número de solicitações que ainda não haviam sido cadastradas, estando presentes em tabelas e anotações, passando tudo para o sistema que utilizava na época.

Levando em consideração o planejamento efetuado, que contou com a inclusão de novos clientes e um novo projeto, novos integrantes serão necessários para a equipe de desenvolvimento. Portanto, um processo organizado passa a ser uma necessidade.

Analisando os objetivos principais da Empresa, a equipe destacou alguns aspectos relevantes para a adoção de uma metodologia de desenvolvimento de software:

- Necessidade de transparência no planejamento e desenvolvimento, com reuniões frequentes com os envolvidos no processo de desenvolvimento para monitorar o progresso;
- O atendimento a mudança constante dos requisitos do projeto;
- A falta de acompanhamento dos problemas identificados no software;
- Horas de trabalho devem ser otimizadas, no sentido de que "trabalhar horas extras" não necessariamente significa "produzir mais".

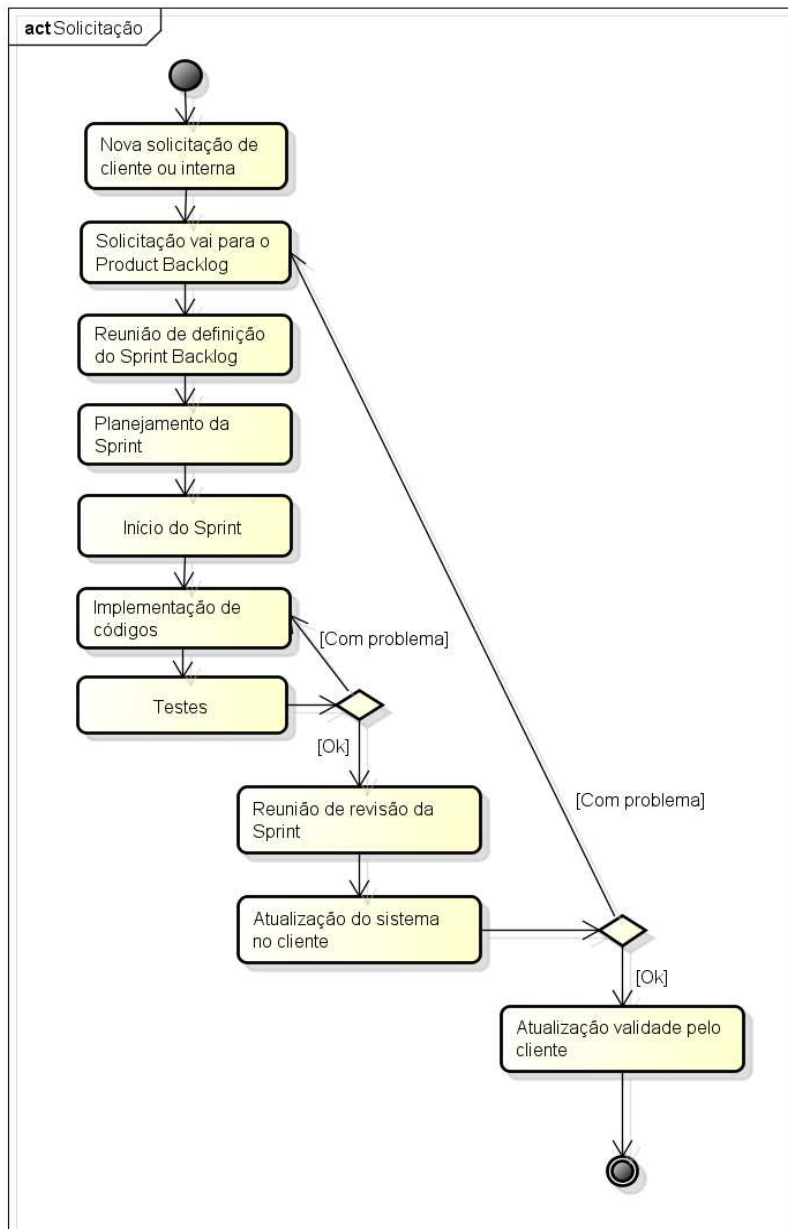
#### 4.3 IMPLANTAÇÃO DO SCRUM

Com base nas características principais apontadas pela equipe realizou-se um estudo de algumas metodologias ágeis e, posteriormente, identificado que o Scrum seria a melhor solução a ser implantada, pois segundo Cohn (2011), é fundamental que seja obtido um reconhecimento de que o processo atual não está dando os resultados esperados, como também, o desejo da adoção do Scrum como ferramenta de solução. Uma apresentação do Scrum foi realizada para

toda a equipe com o objetivo de mostrar suas principais características e o seu funcionamento.

As etapas que seguem descrevem o planejamento para a implantação do Scrum:

1. Organização das solicitações por nível de prioridade em um novo ambiente, que seja mais prático e de acesso mais fácil, produzindo o *backlog* do produto. Alterando também o detalhamento das descrições, adicionando dados mais consistentes, como por exemplo, imagens das telas;
2. Reunião para definição do prazo e do *backlog* para o primeiro *Sprint* a ser realizado;
3. Realização da primeira reunião de encerramento de *Sprint*;
4. Definição da primeira entrega feita aos clientes após a adoção do Scrum.

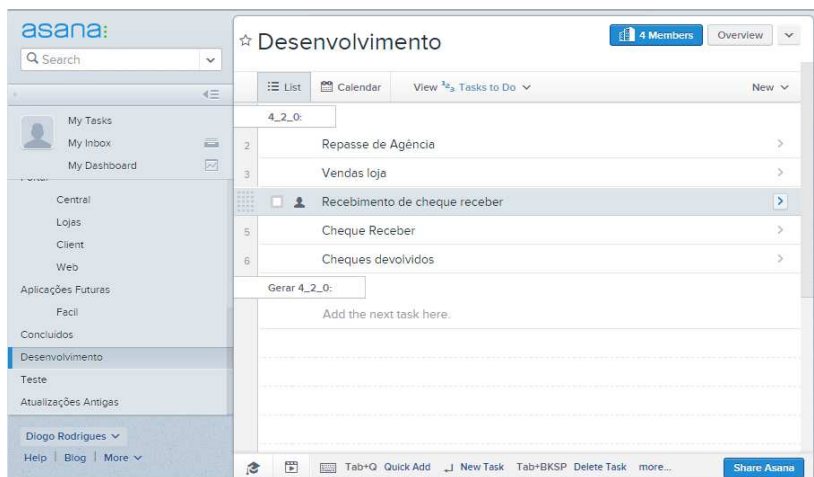


**Figura 7 - Diagrama de atividades da abertura de solicitações após a adoção do Scrum.**

Para realizar o controle do *backlog* do produto, selecionou-se a ferramenta Web Asana. Sendo considerada sólida para gerenciar projetos, possui uma interface amigável e de fácil entendimento, destacando-se também a mobilidade, pois disponibiliza aplicativos móveis para Android e iOS, permitindo que sejam realizadas alterações sem a necessidade de um computador.

A ferramenta possui grande importância para a organização no processo de adoção do Scrum em relação às *Sprints* e ao *backlog* do produto, possibilitando à equipe de desenvolvimento uma visão geral sobre as atividades já realizadas, assim como o acompanhamento das atividades que estão em andamento, bem como as futuras. Possui também a opção de adicionar comentários específicos para cada solicitação, tornando a comunicação entre a programação e os testes muito mais eficiente e simples.

Para melhor organização, as solicitações foram divididas em seções, fazendo parte do *backlog* do produto aquelas referentes aos módulos do sistema e aos clientes. As demais, dizem respeito ao *backlog* da *Sprint* que está sendo desenvolvido, contendo três divisões: o que está no setor de desenvolvimento, o que está no setor de testes e por fim o que já foi concluído, e também a *Sprints* anteriores e atualizações antigas. Como mostra a Figura 8:



**Figura 8 - Ferramenta Asana**

### 4.3.1 Mudança de Papéis

A mudança de papéis em uma equipe pode tornar a adoção do Scrum dolorosa. Sendo com o surgimento de novos papéis ou com a remoção de papéis utilizados anteriormente (COHN, 2011).

A implantação do Scrum trouxe dois novos papéis à equipe: o Scrum *Master* e o dono do produto. Por obter um maior conhecimento sobre o produto e suas funcionalidades o analista de sistemas foi definido como dono do produto. Já o papel de Scrum *Master* ficou a cargo do gerente de projetos e testes, mudando um pouco a sua maneira de trabalhar e se tornando mais flexível e confiante na equipe.

A Tabela 2 mostra as características do papel do gerente de projetos e testes, agora Scrum *Master*, que foi o que mais sofreu alteração nas suas tarefas. Agora a equipe de desenvolvimento divide algumas de suas atividades, com o objetivo de se autogerenciar.

**Tabela 2 - Papéis do Gerente de Projetos e Testes.**

<b>Gerente de Projetos e Testes</b>	<b>Scrum <i>Master</i></b>
Assegura que o que está sendo desenvolvido é realmente o que foi proposto pelo cliente.	É um líder servidor para a equipe auxiliando-a a maximizar o seu rendimento e a adotar o Scrum.
Gerencia todo o projeto, como, riscos, pessoal, custo e compras.	Possui autoridade sobre o processo, mais não sobre a equipe.

O analista de sistemas não teve muito trabalho para se adequar a nova função de dono do produto, pois como a empresa tem uma equipe pequena ele já desempenhava um papel mais próximo desta.

Na equipe Scrum o programador não recebe somente o que deve programar, pois torna-se mais participativo na compreensão dos requisitos do produto. Deve também aceitar tarefas fora de suas áreas específicas quando o objetivo é a conclusão de um *Sprint* (COHN, 2011). A empresa já vinha trabalhando dessa forma, mas não de maneira tão explícita como no Scrum. Já era função do programador realizar os testes básicos no momento da escrita do código e, muitas vezes, está mais próximo do cliente em busca de um melhor entendimento dos requisitos.

Os profissionais de testes também têm uma função diferente com o Scrum, pois assim como o programador não recebe mais as especificações perfeitas, os testadores não tem mais os requisitos perfeitos para garantir as funcionalidades do sistema (COHN, 2011).

Mesmo antes da adoção do Scrum a equipe já mantinha uma boa troca de informação entre todos os integrantes do processo, e após a abordagem essa comunicação ficou ainda mais facilitada. As conversas são frequentes sobre como uma funcionalidade deve ser para gerar qualidade nas entregas.

Ao final de cada *Sprint*, é ideal que seja entregue um software funcionando e com novos incrementos ou correções. Aprender como entregar esse software funcionando é um dos maiores desafios na adoção do Scrum (COHN, 2011). Para obter êxito nesse quesito a empresa buscou controlar as versões de acordo com cada *Sprint*, sendo que no primeiro *Sprint* realizado mudanças pequenas formaram um *Sprint* de duas semanas de duração, gerando um novo incremento do produto.

A coleta de dados é uma tarefa trabalhosa, mas possui ótimos benefícios. Dados positivos estimulam a equipe e combatem o retrocesso, permitindo também quantificar melhor os avanços e os problemas, indicando para onde direcionar as ações de melhoria de processo. Também é muito importante levar em consideração a avaliação de todos os componentes da equipe (COHN, 2011).

#### 4.3 RESULTADOS

Visando quantificar os resultados obtidos após a adoção do Scrum, duas abordagens foram utilizadas. Inicialmente comparou-se os dados obtidos na Tabela 1, que antecedem a implantação do Scrum na empresa, e os dados da Tabela 3, que resultam após a implantação do Scrum.

Posteriormente aplicou-se, um questionário com o intuito de verificar a opinião da equipe sobre toda a mudança no processo de desenvolvimento.

O número de solicitações cadastradas e concluídas, após a adoção do Scrum, é mostrado na Tabela 3:

**Tabela 3 - Solicitações depois da implantação do Scrum.**

<b>Mês / Ano</b>	<b>Solicitações cadastradas</b>	<b>Solicitações concluídas</b>	<b>Período médio para conclusão</b>
Ago / 2014	30	27	52 dias
Set / 2014	56	57	18 dias
Out / 2014	31	28	11 dias
Nov / 2014	40	25	10 dias

<b>Totais</b>	<b>157</b>	<b>137</b>	<b>22 dias</b>
<b>Média mensal</b>	<b>39</b>	<b>34,5</b>	<b>-</b>

Pode se perceber que houve um aumento bastante significativo tanto na média mensal de solicitações cadastradas, tendo 243% de crescimento saltaram de 16 para 39, quanto na média mensal de solicitações atendidas, tendo 460% de crescimento que passaram de 7,5 para 34,5. O Scrum permitiu não só que mais solicitações fossem atendidas, mas fez com que a equipe utilizasse um processo mais rígido de trabalho, sendo assim nenhuma solicitação ficou sem ser cadastrada. Outro aspecto importante a ser verificado é a redução no período médio para conclusão das solicitações, que era de 72 dias e passou para 22 dias.



**Figura 9 - Solicitações concluídas.**

A Figura 9 apresenta o fluxo de solicitações concluídas. Pode-se observar que o mês de setembro apresentou um alto desempenho nesta análise. Ao mostrar para a equipe, foi verificado que houve uma revisão geral de todas as solicitações nesse período, e que muitas delas consideradas rápidas foram resolvidas em um *Sprint* de duas semanas. Sendo que o prazo máximo definido para cada *Sprint* é de um mês.





**Figura 10 - Período médio para a conclusão das solicitações.**

A Figura 10 é referente a média de dias para a conclusão das solicitações. Foram desconsiderados os meses de dezembro de 2013, fevereiro e abril de 2014, pois não apresentaram solicitações concluídas, não obtendo um período médio de conclusão das mesmas. A queda no período médio mostra a evolução da equipe e a importância de se obter um controle mais rígido do *backlog* do produto. Nesses meses entregas mais curtas foram criadas e um maior número de incrementos foram disponibilizados.

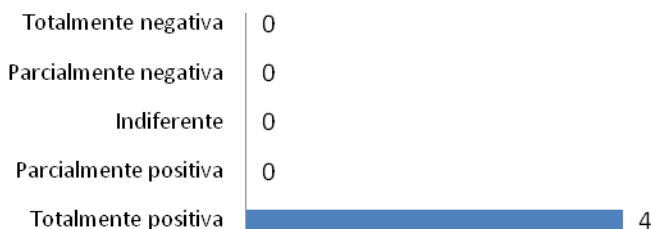
#### 4.3.1 Análise do questionário

Para a avaliação do Scrum não é necessário encontrar resultados e respostas perfeitas, mas sim encontrar resultados e respostas que nos ajudem a responder algumas questões, como por exemplo: “A adoção do Scrum valeu a pena? O que devemos melhorar? Estamos fazendo produtos melhores? Nossos produtos têm menos defeitos? Somos mais rápidos do que costumávamos ser?” (COHN, 2011).

Com o intuito de conhecer melhor a opinião dos envolvidos no processo de adoção do Scrum e responder as perguntas citadas no parágrafo anterior, foi realizado um questionário com os quatro participantes da equipe. Este foi dividido em três partes: a primeira parte traz perguntas em um âmbito mais geral da mudança no processo de desenvolvimento; já a segunda parte traz perguntas referentes ao Scrum, e por fim a terceira parte com perguntas a respeito da visão do processo de desenvolvimento após a adoção do Scrum.

Paul Lawrence (1960), apud Cohn (2011), descreve uma visão interessante sobre mudança, onde Lawrence definiu que a mudança pode ser encarada com dois aspectos diferentes. Um aspecto técnico, que diz respeito à mudança física imposta, e com um aspecto social, que diz respeito à maneira que estas mudanças irão afetar os relacionamentos na empresa. Lawrence define também que os fatos mostram que o aspecto físico é importante, mas que o aspecto social é que irá definir se haverá resistência ou não as mudanças.

Desde a primeira reunião da equipe sobre uma possível mudança foi enfatizada a importância da presença e da opinião de todos, com o objetivo de obter maior aceitação e interesse coletivo. Assim como também as decisões foram tomadas em conjunto, respeitando a área de conhecimento de cada um, mas nivelando todos em um mesmo grau de importância para o sucesso da mudança e da equipe.



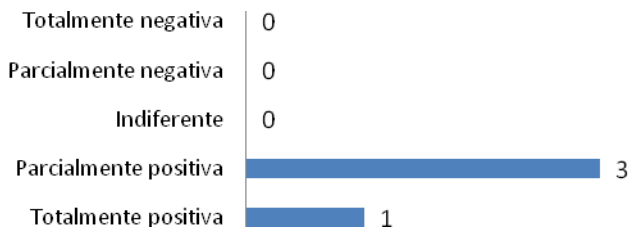
**Figura 11 - Você considera que a mudança do processo de desenvolvimento no geral foi?**



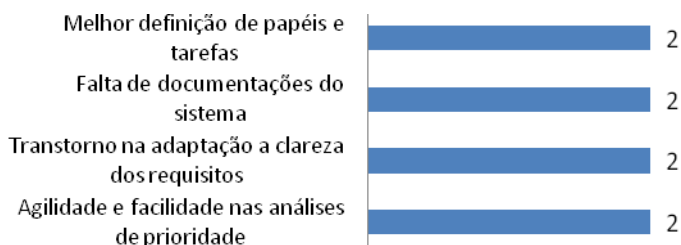
**Figura 12 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior?**

As figuras 11 e 12 mostram os resultados obtidos nas perguntas de cunho geral, em que todos os integrantes definiram como as mudanças são extremamente positivas, destacando-se alguns pontos

positivos, como a organização, agilidade e produtividade, diminuindo um dos maiores problemas encontrados na empresa antes da implantação.

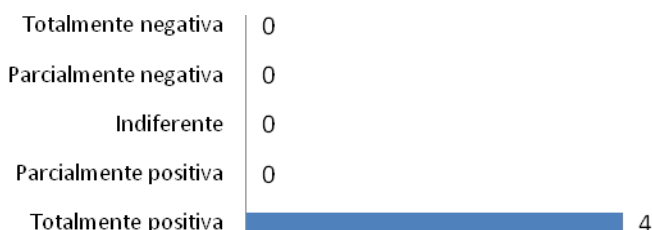


**Figura 13 - Como você avalia a mudança de processo no contexto do gerenciamento de escopo (análise das novas requisições, divisão em tarefas, clareza nos requisitos, documentação)?**



**Figura 14 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior?**

Analisando as figuras 13 e 14 percebeu-se que problemas foram encontrados na documentação do produto e na clareza dos requisitos. Em contrapartida, a definição dos papéis e tarefas a serem realizadas foram muito bem empregadas. O ganho com a organização do *backlog* do produto facilitou muito a análise das solicitações prioritárias. Neste quesito destaca-se a falta de uma documentação mais sólida, sendo que a empresa ainda não a possui e permanece uma oportunidade de melhoria.

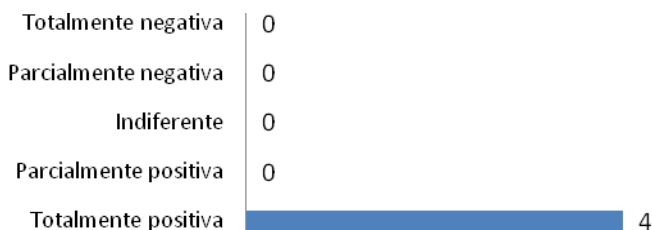


**Figura 15 - Como você avalia a mudança de processo no contexto do gerenciamento de tempo (delegação de tarefas, estimativas de tempo, acompanhamento do desenvolvimento)?**

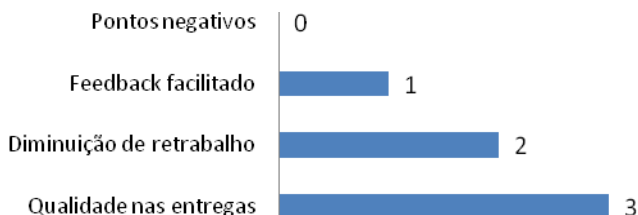


**Figura 16 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior?**

As figuras 15 e 16 mostram o reconhecimento positivo na delegação de tarefas, estimativa de tempo e acompanhamento do desenvolvimento. A transparência das tarefas que estão sendo realizadas foi um ponto de destaque, permitindo um melhor acompanhamento, como também uma melhor definição das versões desenvolvidas, apresentando o que será entregue e quando.

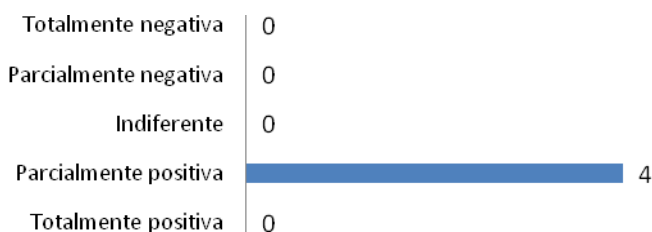


**Figura 17 - Como você avalia a mudança de processo no contexto do gerenciamento de qualidade (validação das implementações, gerenciamento dos releases)?**

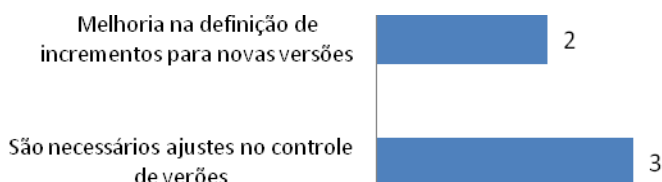


**Figura 18 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior?**

As figuras 17 e 18 mostram a satisfação da equipe quanto à qualidade nas entregas e no gerenciamento destas, tendo em vista que o feedback esperado pelos programadores tem chegado com qualidade, pois os responsáveis pelos testes estão mais próximos e tem uma visão melhor do que realmente é esperado, refletindo em entregas mais consistentes para os clientes.

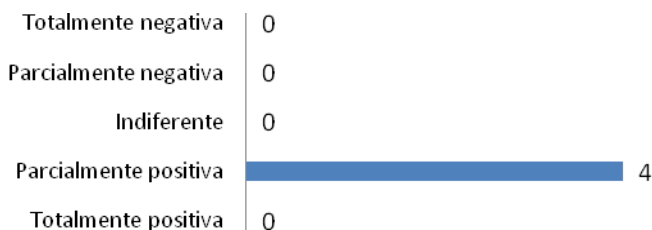


**Figura 19 - Como você avalia a mudança de processo no contexto do gerenciamento de configuração (controle de versões, registro das mudanças, integração contínua)?**



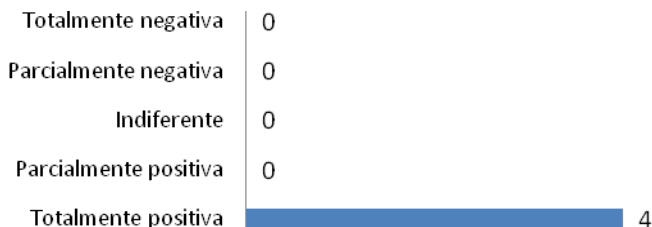
**Figura 20 - Quais seriam os principais pontos a se destacar, conforme seu posicionamento na questão anterior?**

As figuras 19 e 20 mostram que a técnica de controle de versão definida pela empresa não tem atendido as necessidades de acompanhamento e controle de versões. Por outro lado, cada nova versão possui incrementos bem definidos. É necessário uma melhor avaliação dos pontos a serem melhorados que foram identificados pela pesquisa.



**Figura 21 - Como você avalia o processo de Implantação do Scrum na empresa?**

A Figura 21 apresenta a satisfação parcialmente positiva se tratando do processo de implantação. Alguns fatores podem ter sido provenientes de problemas enfrentados pela empresa há um bom tempo, tais como, falta de uma descrição mais abrangente das solicitações e o controle das mesmas. Reunir dados antes da implantação do Scrum foi uma tarefa difícil, pois a organização era uma das grandes falhas encontradas na empresa.



**Figura 22 - Como você avalia a comunicação entre os membros da equipe após a implantação do Scrum?**

A Figura 22 mostra a satisfação totalmente positiva após a adoção do Scrum em relação à comunicação entre a equipe. Destacando a ferramenta Asana como maior facilitador dessa comunicação, pois permite a equipe trocar mensagens sobre uma solicitação.

## 5 CONSIDERAÇÕES FINAIS

As metodologias tradicionais não têm atendido as necessidades dos projetos de software, que exigem cada vez mais uma resposta rápida e eficiente à demanda dos clientes, além da alta competitividade. As metodologias ágeis vieram suprir estas exigências garantindo uma formalização e uma produção de alto desempenho, sem deixar de lado o atendimento aos requisitos de segurança e controle de todo o processo de desenvolvimento de software.

A adoção do Scrum na Empresa despertou grande interesse da equipe devido ao potencial desta metodologia em melhorar os processos organizacionais. O impacto das mudanças foi positivo, fazendo com que a equipe trabalhasse em conjunto, melhorando a autoestima e assim o rendimento de todos os envolvidos no processo. Um ponto a se destacar foi à quantificação dos resultados obtidos, tanto com o levantamento de dados antes e depois da adoção, quanto com o questionário apresentado no capítulo anterior.

O questionário foi de extrema importância, permitindo chegar a conclusão de qual a opinião da equipe, e juntamente com as métricas obtidas, dar um *feedback* adequado a mesma.

Podem-se apresentar algumas contribuições para a empresa, como por exemplo: organização e controle de solicitações, bem como comunicação otimizada e análise do processo através de métricas. Foram identificadas algumas áreas onde alterações precisam ser revistas ou adotadas, visando melhorar o procedimento atual, como por exemplo a documentação do software e um controle padronizado para versões, tarefas essas que no Scrum não abrange.

Ressalta-se que a mudança é principalmente cultural dentro de uma empresa, uma vez que todo o processo é baseado na organização de tarefas bem definidas, e todos devem segui-las.

Exalta-se como critério chave para o sucesso o envolvimento de toda a equipe no processo. Não havendo rejeição inicial a proposta, sendo uma decisão tomada em conjunto.

Ao final deste trabalho conclui-se que os objetivos propostos foram plenamente alcançados. Inicialmente, foi realizado um estudo da metodologia ágil Scrum e após obter esse conhecimento, foi feito um planejamento para adaptação da empresa, realizando todas as mudanças necessárias, e por fim, apresentado uma análise de todo o processo.



## REFERÊNCIAS

ALBINO, Raphael Donaire; SOUZA, Cesar Alexandre de; PRADO, Edmir Parada Vasques. **Benefícios alcançados através de um modelo de Gestão Ágil de Projeto em uma empresa de jogos eletrônicos**. 2013. 15 f. TCC (Graduação) - Curso de Administração, Usp - Universidade de São Paulo, São Paulo, 2013.

COHN, Mike. **Desenvolvimento de Software com Scrum**: Aplicando métodos ágeis com sucesso. Porto Alegre: Bookman Companhia Editora Ltda., 2011.

GONÇALVES, Bruno Leonardo. **Engenharia de Software**. Disponível em: <<https://sites.google.com/site/brunolmfg/engenharia-de-software>>. Acesso em: 10 set. 2014.

HIRAMA, Kechi. **Engenharia de Software**: Qualidade e Produtividade com Tecnologia. Rio de Janeiro: Elsevier Editora Ltda., 2012.

HOUAISS, Antônio; VILLAR, Mauro de Salles; FRANCO, Francisco Manoel de Mello. Houaiss: **Dicionário da língua portuguesa**. Rio de Janeiro: Editora Objetiva Ltda., 2001.

JUBILEU, Andrea Padovan. **Modelo de Gestão do Processo de Venda e Desenvolvimento de Software On-Demand para MPE's**. 2008. 330 f. Tese (Doutorado) - Curso de Engenharia de Produção, Universidade Federal de São Carlos (ufscar), São Carlos, 2008.

LANDIM, Henrique Farias. **Uma Abordagem de Monitoramento dos Fatores e Condições que influenciam nas Práticas Ágeis**. 2012. 185 f. Dissertação (Mestrado) - Curso de Mestrado em Informática Aplicada, Fundação Edson Queiroz Universidade de Fortaleza Centro de Ciências Tecnológicas, Fortaleza, 2012.

LEITÃO, Michele de Vasconcelos. **Aplicação de Scrum em Ambiente de Desenvolvimento de Software Educativo**. 2010. 72 f. TCC (Graduação) - Curso de Engenharia da Computação, Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, 2010.

LEITE, Alessandro Ferreira. **Metodologia de desenvolvimento de Software**. Disponível em: <<http://www.devmedia.com.br/metodologia-de-desenvolvimento-de-software/1903>>. Acesso em: 02 set. 2014.

WIRTH, Niklaus. **Uma breve História da Engenharia de Software**. 2012. Disponível em: <[http://marathoncode.blogspot.com.br/2012/07/uma-breve-historia-da-engenharia-de\\_10.html](http://marathoncode.blogspot.com.br/2012/07/uma-breve-historia-da-engenharia-de_10.html)>. Acesso em: 25 set. 2014.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software: Fundamentos, Métodos e Padrões**. 3. ed. Rio de Janeiro: Ltc - Livros Técnicos e Científicos Editorial S.a., 2009.

PRESSMAN, Roger S.. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 2010.

RAMOS, Almir et al. **Engenharia de Software: Modelo Cascata**. Disponível em: <<http://modelocascata.blogspot.com.br/>>. Acesso em: 01 set. 2014.

ROCHA, Igor. **Lean Software Development: Desenvolvimento de Software Leve**. 2013. Disponível em: <<http://igorrocha.com.br/lean-software-development/>>. Acesso em: 05 set. 2014.

SANTOS, Renata. **Metodologias Ágeis para desenvolvimento de software**. 2013. Disponível em: <<http://www.blogti.microcampsp.com.br/metodologias-ageis-para-desenvolvimento-de-software-parte-i/>>. Acesso em: 04 set. 2014.

**SCRUM**. Disponível em: <<http://desenvolvimentoagil.com.br/scrum/>>. Acesso em: 08 out. 2014.

SILVA, Carlos Eduardo Azevedo Costinhas da. **Um estudo de caso sobre a adoção de práticas ágeis em um ambiente tradicional**. 2013. 103 f. TCC (Graduação) - Curso de Sistemas de Informação, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

STEFFEN, Juliana Berossa. **Lean para desenvolvimento de Software**. Afinal, o que é isto? 2011. Disponível em: <[https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/lean\\_para\\_desenvolvimento\\_de\\_sw\\_o\\_que\\_\\_c3\\_a9\\_isso\\_afinal12?lang=en](https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/lean_para_desenvolvimento_de_sw_o_que__c3_a9_isso_afinal12?lang=en)>. Acesso em: 04 set. 2014.

SCHWABER, Ken. **Agile Project Management with Scrum**. Redmond, Washington: Microsoft Press, 2004. 175 p.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game**. 2013. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>>. Acesso em: 02 nov. 2014.

VARASCHIM, Jacques Douglas. **Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet**. 2009. Curso de Ciência da Computação, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

**APÊNDICE A – Questionário aplicado para levantar a opinião dos envolvidos sobre o processo de adoção do Scrum.**

## Questionário de avaliação

Nome: \_\_\_\_\_

Cargo: \_\_\_\_\_

### 1. Geral

1.1 Você considera que a mudança de processo de desenvolvimento foi, no geral:

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa

1.2 Quais seriam os principais pontos (positivos/negativos) a se destacar, conforme seu posicionamento na questão anterior? (dissertativa)

### 2. Scrum

2.1 Como você avalia a mudança de processo no contexto do gerenciamento de escopo (análise das novas requisições, divisão em tarefas, clareza nos requisitos, documentação)?

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa

2.2 Quais seriam os principais pontos (positivos/negativos) a se destacar, conforme seu posicionamento na questão anterior? (dissertativa)

2.3 Como você avalia a mudança de processo no contexto do gerenciamento de tempo (delegação de tarefas, estimativas de tempo, acompanhamento do desenvolvimento)?

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa

2.4 Quais seriam os principais pontos (positivos/negativos) a se destacar, conforme seu posicionamento na questão anterior? (dissertativa)

2.5 Como você avalia a mudança de processo no contexto do gerenciamento de qualidade (validação das implementações, gerenciamento dos releases)?

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa

2.6 Quais seriam os principais pontos (positivos/negativos) a se destacar, conforme seu posicionamento na questão anterior? (dissertativa)

2.7 Como você avalia a mudança de processo no contexto do gerenciamento de configuração (controle de versões, registro das mudanças, integração contínua)?

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa

2.8 Quais seriam os principais pontos (positivos/negativos) a se destacar, conforme seu posicionamento na questão anterior? (dissertativa)

### 3. Específica

3.1 Como você avalia o processo de Implantação do Scrum na empresa?

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa

3.2 Como você avalia a comunicação entre os membros da equipe após a implantação do Scrum?

- ☐ Totalmente positiva
- ☐ Parcialmente positiva
- ☐ Indiferente
- ☐ Parcialmente negativa
- ☐ Totalmente negativa